

Modul dispečerské plachty pro dopravní monitorovací systém

Dispatching Board for Traffic Monitoring System

Zadání bakalářské práce

Student: **Tomáš Jurásek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Modul dispečerské plachty pro dopravní monitorovací systém
Dispatching Board for Traffic Monitoring System**

Zásady pro vypracování:

Cílem bakalářské práce je vytvoření modulu dispečerské plachty pro dopravní monitorovací systém vyvíjený společností GX Solutions ve spolupráci s VŠB-TU Ostrava. Tento modul bude určen pro manuální plánování přeprav pomocí k tomu vytvořeného speciálního uživatelského rozhraní. Tento modul bude řešit mimo jiné také provázání s modulem optimalizace přeprav. Při vytváření bakalářské práce student obdrží základní knihovny pro komunikaci s databází a dalšími existujícími částmi systému.

Jednotlivé body bakalářské práce:

1. Analýza současného stavu v řešené problematice a porovnání stávajícího řešení firmy GX Solutions s dalšími existujícími softwary.
2. Implementace databázové vrstvy pro uložení stavů aktuálního zobrazení dispečerské plachty pro jednotlivé uživatele.
3. Rozšíření existujícího řešení specializovaného kalendáře pro manuální plánování přeprav.
4. Testování vytvořeného modulu s využitím reálných dat.

Seznam doporučené odborné literatury:

- [1] GX Solutions. Manuál na obsluhu SW Truck Data Memory 2010 v2.2
[2] PostgreSQL: <http://www.postgresql.org>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Kateřina Slaninová, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2015

.....*Juráček*.....

V první řadě bych rád poděkoval Ing. Kateřině Slaninové, Ph.D. za vedení bakalářské práce. Dále bych rád poděkoval Ing. Marku Boberovi z firmy GX Solutions Bohemia s.r.o., za cenné rady a konzultace při vývoji aplikace. Taky bych chtěl poděkovat své přítelkyni a rodině za podporu při studiu.

Abstrakt

Bakalářská práce „Modul dispečerské plachty pro dopravní monitorovací systém“ se zabývá vytvořením nového modulu u vznikajícího systému pro dopravu a logistiku, který vyvíjí firma Gx Solutions. Systém je zaměřen na GPS sledování, monitoring vozidel, hlídání a vyhodnocování stylu jízd řidičů a plánování přeprav. Cílem této práce je navrhnout a vytvořit modul dispečerské plachty, který se stará o celý proces vytvoření přeprav a následnou manipulaci s nimi. Práce je tvořena pomocí programovacího jazyka C#, frameworku ASP.NET MVC, skriptovacího jazyka JavaScript a klientských komponent DevExpress.

Klíčová slova: Gx Solutions, dispečerská plachta, přeprava, zastávky, konfigurace přeprav, c#, devexpress, logistika a doprava, systém pro dopravu

Abstract

Bachelor thesis "Dispatching Board for Traffic Monitoring System" deals with the creation of a new module for the nascent transport and logistics system being developed by the company Gx Solutions. The system is focused on GPS tracking, vehicle monitoring, monitoring and evaluation of driver's style and transport planning. The aim of this thesis is to design and implement a module for a dispatcher board, which will deal with the whole process for creating a shipment and its subsequent manipulation. The thesis is being created using the C# programming language, ASP.NET MVC framework, JavaScript scripting language and DevExpress client components.

Keywords: GX Solutions, dispatcher board, transport, stops, transport settings, c#, devexpress, transport and logistics, system for transport

Seznam použitých zkratk a symbolů

GPS	– Global Positions System
C#	– C Sharp
WCF	– Windows Communication Foundation
MVC	– Model-View-Controller
ER Diagram	– Entity Relationship Diagram
GX Solutions	– Global Monitoring Experts

Obsah

1	Úvod	5
2	Gx Solutions Bohemia s.r.o	6
2.1	Profil společnosti	6
3	Analýza současného stavu	8
3.1	Stávající softwarové řešení firmy	8
3.2	Softwarové řešení konkurence	11
4	Návrh	14
4.1	Architektura projektu	14
4.2	Návrh modulu	15
4.3	Databázová vrstva modulu	21
5	Implementace	25
5.1	Výkonné jádro modulu	25
5.2	Testování a refaktoring modulu	32
6	Závěr	34
7	Reference	35
	Přílohy	36
A	Dispečerská plachta	37
B	Databáze	45

Seznam tabulek

1	Tabulka porovnání produktů	13
2	Datový slovník tabulky TransportConfig	45
3	Datový slovník tabulky TransportConfigData	45
4	Datový slovník tabulky TransportStopConfigData	45
5	Datový slovník tabulky TransportData	46
6	Datový slovník tabulky TransportDataRow	46
7	Datový slovník tabulky TransportStopDataRow	46
8	Datový slovník tabulky TransportStopDataColumn	46
9	Datový slovník tabulky TransportStopPoint	46

Seznam obrázků

1	Přehled řešení [7]	7
2	Konfigurace přeprav	9
3	Vygenerovaný formulář pro přepravu	10
4	Formulář pro zastávku	11
5	Dispečerská plachta	11
6	Diagram komponent	15
7	Případ užití	16
8	Návrh konfigurace přeprav - nastavení	17
9	Wireframe - konfigurace přeprav	18
10	Wireframe - nová přeprava	20
11	Wireframe - zastávky	20
12	ER diagram - konfigurace přeprav	22
13	ER diagram - přeprava	24
14	Webdispečink - Jednoduchá přeprava	37
15	Webdispečink - Proces vytvoření přepravy	38
16	Webdispečink - Dispečerská plachta	39
17	Webdispečink - Seznam přeprav	40
18	Doprava 3K - Nová přeprava	41
19	Doprava 3K - Dispečerská plachta	41
20	Vytvořené uživatelské rozhraní konfigurace přeprav	42
21	Vytvořené uživatelské rozhraní pro přepravu	43
22	Vytvořené uživatelské rozhraní pro vizuální dispečerskou plachtu	44

Seznam výpisů zdrojového kódu

1	Algoritmus pro získání hodnot k položkám	27
2	Algoritmus pro namapování hodnot na konkrétní řádek zastávek	29
3	Algoritmus pro namapování hodnot na konkrétní řádek zastávek	30
4	Metoda, která nastaví mapování třídy pro přepravu na položku komponenty	31
5	Metoda, která nastaví mapování třídy vozidla (resource) na zdroj pro kom- ponentu	31
6	Metoda získá seznam upravených přeprav	32
7	Metoda vracející seznam přeprav podle data a vozidla	32

1 Úvod

Cílem této práce je vyvinout nový modul, který bude součástí vznikajícího systému pro dopravu a logistiku, který vyvíjí firma Gx Solutions. Firma Gx Solutions nabízí své řešení firmám, které potřebují spravovat a řídit svůj vozový park. Systém obsahuje funkce jako např. GPS sledování, vyhodnocení jízd řidičů, plánování přeprav (objednávek), tras a jiné. Z důvodu velké komplexnosti je systém rozdělen na moduly. Mým cílem je navrhnout a vytvořit modul dispečerské plachty, který má za úkol vytváření přeprav (objednávek) na základě nastavení modulu a následné grafické zobrazení přeprav a vozidel. Prvním krokem je vytvoření speciálního nastavení modulu, které určuje povinné vstupy do formuláře přeprav a zastávek. Na základě tohoto nastavení vygenerujeme dynamické formuláře pro přepravu a zastávky přepravy. K zastávkám dále můžeme vytvářet průjezdné body, které budeme moci vybrat na mapě. Uložené přepravy se budou zobrazovat ve speciálním uživatelském rozhraní, které bude zobrazovat vytvořené přepravy k jednotlivým vozidlům. V tomhle rozhraní bude dispečerovi umožněno upravovat vytíženost jednotlivých vozidel a jednoduše s přepravami manipulovat. Pro snadnější pochopení modulu lze využít diagram případu užití v kapitole 4.2 na obrázku 7.

Druhá kapitola práce se zabývá analýzou modulu dispečerské plachty současného řešení firmy Gx Solutions a produkty konkurence, které poskytují volný demo přístup do svých aplikací. Tato kapitola nás blíže seznámí s funkcionalitou a problematikou modulu.

Ve třetí kapitole navrhne uživatelské rozhraní a požadovanou funkčnost modulu. Celý modul se dále dělí na menší logické části a je potřeba správně navrhnout propojení mezi nimi. Podle navržené funkčnosti je třeba vytvořit databázovou strukturu pro uložení dat.

V kapitole čtyři budeme provádět implementaci modulu na základě jeho návrhu. Modul potom napojíme na databázovou strukturu. Na konci kapitoly shrneme způsoby testování modulu a případně opravíme nalezené chyby.

V závěru vyhodnotíme poznatky, které jsme získali při vývoji modulu, a problémy, které se vyskytly během vývoje.

2 Gx Solutions Bohemia s.r.o

2.1 Profil společnosti

Firma GX Solutions Bohemia s.r.o. vznikla v roce 1998. Společnost poskytuje služby firmám, které podnikají v oblasti logistiky a dopravy. Mezi první produkty firmy patřilo jen řešení pro monitorování vozidel. Později se nabídka rozrostla o komplexnější systémy, díky kterým zákazník dokáže kontrolovat stav a výkon vozidla, komunikaci s řidiči a využívat dalších možností pro efektivnější práci. Díky softwaru postavenému na modulech může firma nabídnout zákazníkům software ušitý na míru. Firma poskytuje několik řešení, která obsahují funkce z vybraných oblastí [1]:

- Doprava a logistika - řešení nabízí monitoring vozidel, plánování tras, kontrolu pohonných hmot a diety řidiče. Na základě jízd je možné vytvořit užitečné reporty o stylu jízdy a alarmy pro dispečink. Díky komunikačnímu systému může dispečer neustále komunikovat s řidičem [2].
 - Gx Bus - autobusová doprava
 - Gx Truck - nákladní a přepravní doprava
 - Gx Loco - lokomotivy
- Výroba a průmysl - řešení je vhodné pro zásobování a velké sklady. Jeho hlavním úkolem je zjistit neefektivní využívání technik a tím zamezit finančním ztrátám a klást důraz na bezpečnost procesů [3].
 - Gx Store - vysokozdvížné a plošinové vozíky
 - Gx Personal - sledování osob a zvířat
- Obchod a služby - online sledování vozidel a zaznamenávání jízd do tzv. knihy jízd pro pozdější tvorbu reportů. Důraz je kladen na bezpečnost vozidel [4].
 - Gx Fleet - firemní vozové parky a taxislužby
- Stavebnictví a zemědělství - hlavní uplatnění řešení ve stavebnictví a těžebních pracích. Je možné vozidla kontrolovat online, kontrolovat práci a efektivitu vozidel a napomáhat zefektivnit jejich práci [5].
 - Gx Build - stavební a technické stroje
 - Gx Agro - zemědělské stroje na těžbu dřeva a jiné
- Technické a komunální služby - monitoring komunálních vozidel, monitoring správy a údržby komunikací. Dokáže kontrolovat kvalitu služeb kvůli nespokojenosti občanů v obcích či městech [6].
 - Gx Minicipal - komunální a technické služby

Z důvodu velké komplexnosti systému firma vytvořila několik řešení, která obsahují vybrané moduly a funkce z dané oblasti. Jedná se o tzv. produktovou řadu, která je zobrazena na obrázku 1.

Přehled řešení
Komplexní monitoring vašeho vozového parku

Legenda:

- standardní možnosti
- rozšířené možnosti
- obsahuje
- neobsahuje
- může obsahovat

	Měření paliva	Identifikace návěsů / kontejnerů	Online stav a poloha	CAN Bus	Identifikace řidiče	Evidence řidičů a pojistných událostí	Alarm / Panik	Management tras	Otáčky motoru	Management přeprav	Tachograf	Integrace s jiným IS	Perifere	Komplexní záznam provozu	Online komunikace s řidičem	Sledování přidaných zařízení
fleet osobní doprava, firemní vozové parky	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
truck nákladní a kamiónová doprava	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
build stavební stroje a zařízení	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
municipal komunální a technické služby	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
store manipulační technika	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bus autobusová doprava	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
logistic distribuční a spediční společnosti	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
loco železniční a kolejová doprava	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
agro zemědělské stroje a zařízení	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
personal monitoring osob a zvířat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Obrázek 1: Přehled řešení [7]

3 Analýza současného stavu

Dříve než se budeme zabývat modulem dispečerské plachty, je potřeba se seznámit se základními pojmy modulu.

Dispečerská plachta

Jedná se o jeden z důležitých modulů v komplexním informačním systému. Modul můžeme chápat také jako plugin, nebo část systému. Tenhle modul má na starosti celý proces spojený s přepravou. Jak už vytvoření přepravy a její komponenty, tak pozdější manipulace a správa přepravy.

Konfigurace přeprav

Řeší nastavení celého modulu dispečerské plachty. Většina formulářů v modulu je nastavitelná zákazníkem a zákazník má možnost vytvořit požadované položky ve formulářích v nastavení konfigurace přeprav.

Přeprava

Přepravu lze chápat jako konkrétní objednávku, kterou zákazník vytvoří. Ta vzniká pomocí vygenerovaných formulářů na základě konfigurace přeprav. Přeprava může obsahovat několik zastávek, které dále zahrnují několik průjezdných bodů.

Vizuální dispečerská plachta

Speciální uživatelské rozhraní (upravený kalendář) slouží pro grafické zobrazení vytvořených přeprav. Přepravy a vozidla zobrazené na rozhraní je možné filtrovat na základě určitých kritérií. S přepravou můžeme dále různě manipulovat a měnit její hodnoty.

3.1 Stávající softwarové řešení firmy

Firma nabízí desktopovou aplikaci, která se jmenuje Smart TDM [8]. Jedná se o aplikaci postavenou na modulech. V rámci modulů pak firma nabízí různá řešení, která jsou zobrazena na obrázku 1. Cílem této práce ale není analyzovat produkt jako celek, ale jen modul s názvem dispečerská plachta.

Většina komponent v modulu jde nastavit přímo zákazníkem. Od tohoto nastavení se odvíjejí položky ve formulářích přeprav a zastávek. Dříve než zákazník začne s modulem dispečerské plachty intenzivně pracovat, musí provést toto nastavení. Konfiguraci přeprav můžete vidět na obrázku 2.

Obrázek 2: Konfigurace přeprav

Konfigurace přeprav zahrnuje nastavení:

- Přeprav - zákazník si vytvoří položky (vstupy), které budou vygenerovány u formuláře při tvorbě nové přepravy.
- Zastávek - totožné nastavení jako u přeprav, jen vytvořené položky spadají pod formulář zastávek.
- Obecné nastavení - pro nastavení ostatních parametrů v rámci modulu.

Při tvorbě nových položek si můžeme vybrat z několika předem naprogramovaných typů. Každý typ lze dále specifičtěji formátovat. Zde je uveden seznam typů položek a konkrétních formátů:

- Textový
- Časový
 - hodiny
 - datum
- Číselný
 - celé číslo
 - číslo s desetinnou čárkou
- Objektový
 - druh nákladu
 - typ zastávky
 - řidič
 - zákazník

U položky musíme dále zvolit:

- Aktivní – položka se bude zobrazovat ve formuláři
- Povinná - položka bude povinná k vyplnění ve formuláři
- Zobrazit v tabulce – položka se bude zobrazovat v tabulce přeprav
- Odeslat do terminálu – systém odešle hodnotu do terminálu řidiče
- Pozice položky – určuje pořadí položky ve formuláři

Pokud si zákazník v konfiguraci přeprav označí možnost použít zastávky, je potřeba provést toto nastavení taky pro zastávky. Na základě nastavení se dále odvíjí zobrazení formulářů pro přepravy a zastávky (obr. 3). V rámci jedné přepravy je možné vytvořit

The screenshot displays a software interface for managing transport orders. On the left, there is a form with the following fields and controls:

- Stav:** A dropdown menu set to 'Vytvořená'.
- Číslo přepravy:** A text input field.
- Datum založení:** A date selection dropdown.
- Vozidlo:** A text input field.
- ID přepravy:** A text input field.
- Datum přepravy:** A date selection dropdown.
- Cena:** A text input field.
- Zákazník:** A dropdown menu with '<nezadáno>' selected.
- Poznámka:** A text input field.
- Poznámka disp.:** A text input field.
- Odeslat do vozidla:** A checkbox.

Below these fields, it shows 'Celková vzdálenost: 0 m, Celková doba: 00:00'. At the bottom of the form are four buttons: 'Přiřadit vybrané vozidlo', 'Odeslat nyní', 'Odeslat pozděj', and 'Stornovat'.

On the right side of the form is a map of Central Europe, showing cities like Praha, Brno, Olomouc, and Bratislava. A scale bar indicates 50 km. To the right of the map are several map controls: 'Standardní', 'Alternativní', 'Satelitní', 'Terénní', a zoom in (+) button, a zoom out (-) button, a 'gb' button, a 'firma' button, a 'švihanka' button, and a 'Vzorový výřez' button.

Below the map is a table titled 'Zastávky' (Stops) with the following columns: 'Poradí', 'Typ zastávky', 'Datum', 'Stát, PSČ', 'Druh nákladu', 'Vzdálenost', 'Doba', and 'Stav'. The table is currently empty. To the right of the table are three vertical buttons: '+', '-', and a double-headed arrow.

At the very bottom of the interface is a toolbar with icons and labels for 'Nový', 'Změnit', 'Uložit', 'Storno', 'Smazat', 'Aktualizovat', 'Předchozí', and 'Další'.

Obrázek 3: Vygenerovaný formulář pro přepravu

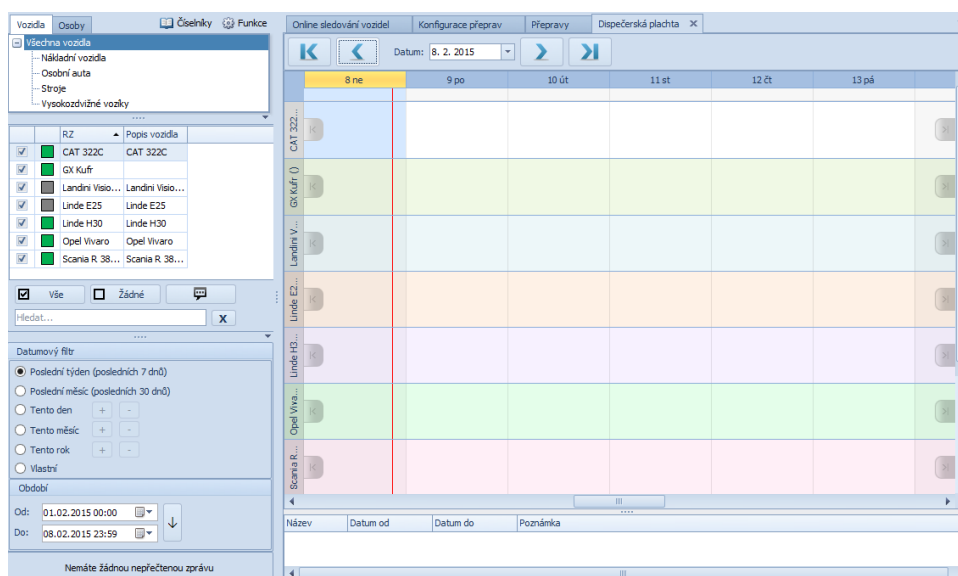
několik zastávek. Jejich počet není omezen. Při vytváření zastávky (obr. 4) je vždy položka Cílový bod povinná a musí být vyplněna. Je možné ji vyplnit pravým kliknutím do mapy a přiřadit bod. Až při uložení celé přepravy můžeme danou přepravu přiřadit vybranému vozidlu a odeslat řidiči do terminálu, který se nachází ve vozidle.

Formulář pro zastávku s následujícími poli:

- Cílový bod: <nezadáno>
- Typ zastávky: <nezadáno>
- ID zastávky: [prázdné]
- Datum: [prázdné]
- Stát, PSČ: [prázdné]
- Město, Ulice: [prázdné]
- Druh nákladu: <nezadáno>
- Množství: [prázdné]
- Kontakt: [prázdné]
- Tel. číslo: [prázdné]
- Poznámka: [prázdné]

Obrázek 4: Formulář pro zastávku

Na obrázku 5 můžeme vidět prázdnou dispečerskou plachtu, ve které by se měly zobrazovat již vytvořené přepravy, se kterými by se dalo manipulovat a tím řídit vytíženost vozidel. Z technických důvodů tato část modulu nefunguje správně a není možné s dispečerskou plachtou pracovat.



Obrázek 5: Dispečerská plachta

3.2 Softwarové řešení konkurence

Modul dispečerské plachty není obvyklá záležitost a mnoho firem tento modul nenabízí s vizuálním zobrazením přeprav. Z tohoto pohledu byl problém nalézt firmy, které nabí-

zejí celý modul a také poskytují demo přístup do své aplikace. U firem, které nenabízejí demo přístup, jsme čerpali z informací dostupných na webových stránkách firmy, nebo reklamních letáčích.

3.2.1 Webdispečink

Produkt Webdispečink je vyvíjen firmou Hi Software development s.r.o. Jde o komplexní systém pro řízení vozového parku a jeho GPS sledování [9]. V systému se modul dispečerské plachty jmenuje Spedice. Formuláře pro přepravu jsou programově předdefinované a není možnost je doplnit o určitou položku. Přeprava se reprezentuje položkami: Název, Vozidlo, Datum od, Datum do, Odkud, Kam, Zastávky, Km, Stav, Odjezd, Příjezd. Firma nabízí dva postupy, jak vytvořit přepravu:

- První postup je vhodný pro stavební firmy, které nepotřebují k přepravě zastávky a jejichž cílem je přepravit vozidla z bodu A do bodu B. Tvorba nové přepravy se provádí přímo na vizuální dispečerské plachtě, kde stačí zvolit Vytvořit záznam. Jednoduchý formulář (Příloha A obr. 14) obsahuje jen podstatné informace, nákladku a vykládku. Vizuální dispečerská plachta není dále provázána s žádnou částí aplikace.
- Další postup je komplikovanější a je vhodný pro nákladní dopravu a jiné společnosti, které potřebují využívat zastávky a podrobnější informace v rámci přepravy. Vytvoření nové přepravy se provádí v záložce Spedice, kde formulář obsahuje několik záložek (Příloha A obr. 15).
 - Nová přeprava - vyplníme základní informace jako název přepravy, vozidlo, datum a konec přepravy a poznámku.
 - Zastávky - vytvoříme zastávky, kterými bude vozidlo projíždět. Maximální počet zastávek, které můžeme vytvořit v přepravě je 8.
 - Mapa - na základě vytvořených zastávek se vykreslí trasa. Tu lze dále upravovat.
 - Alarmy - tato záložka umožňuje vytvořit alarmy na překročení teploty nebo vybočení z trasy o několik kilometrů.
 - Jízdy - zaznamenává jízdu vozidla a zobrazuje ujetou vzdálenost.
 - Provedené akce - záložka umožňuje vytvořit akci, kterou musí řidič na zastávce splnit, např. odeslat informaci o nákladce.
 - Fakturace - na základě vyplněných hodnot vypočítá zisk z dané přepravy.

Po vytvoření přepravy můžeme zobrazit seznam přeprav dvěma způsoby. Prvním způsobem je zobrazení na vizuální dispečerské plachtě (Příloha A obr. 16), kde je můžeme dále upravovat. Druhým způsobem je tabulkové zobrazení (Příloha A obr. 17) Na vizuální dispečerské plachtě je možné filtrovat zobrazená data [10].

3.2.2 Doprava 3K

Doprava 3K je informační systém vyvíjený firmou KSH-Data s.r.o. Software je postaven na modulovém řešení. Zákazník si dále může dokoupit jednotlivé moduly podle svých potřeb. Firma nabízí také modul Spedice, který řeší celý proces od přijetí objednávky až po konečnou kalkulaci. Součástí tohoto modulu je i vizuální dispečerská plachta, která graficky zobrazuje vytíženost vozidel [11]. Firma nenabízí přístup k demo aplikaci svého řešení. Na obrázcích 18 a 19 v Příloze A můžeme vidět uživatelské rozhraní pro vytvoření přepravy a její zobrazení v dispečerské plachtě.

3.2.3 Shrnutí

Analýza měla za úkol nás blíže seznámit s modulem dispečerské plachty pro dopravu a logistiku, které jsou na trhu dostupné. Z důvodu malé dostupnosti jsme byli schopni vyzkoušet produkty jen od dvou firem. Jednalo se o firmu Gx Solutions a její aplikaci smartTDM a firmu Hi Software development a jejich systémem Webdispečink. Na základě těchto analýz vznikla jednoduchá tabulka pro srovnání produktů.

	Gx Solutions s.r.o.	Hi Software development s.r.o.
Možnosti si modul nastavit	Ano	Ne
Použít zastávky v přepravě	Ano	Ano
Funkční vizuální dispečerská plachta	Ne	Ano
Filtrování přeprav	Ne	Ano
Drag and drop na DP	Ne	Ne
Editace přeprav na DP	Ne	Ano
Možnosti tisku detailu přepravy v DP	Ne	Ano
Ovladatelnost	Složitá	Jednoduchá
Grafické zobrazení	Moderní	Zastaralé

Tabulka 1: Tabulka porovnání produktů

4 Návrh

U návrhu nového produktu se klade důraz na multiplatformnost. Bude se jednat o webového klienta a později taky mobilního a desktopového klienta. Systém jako celek by měl zvládat velký nápor příchozích dat a nápor od uživatelů.

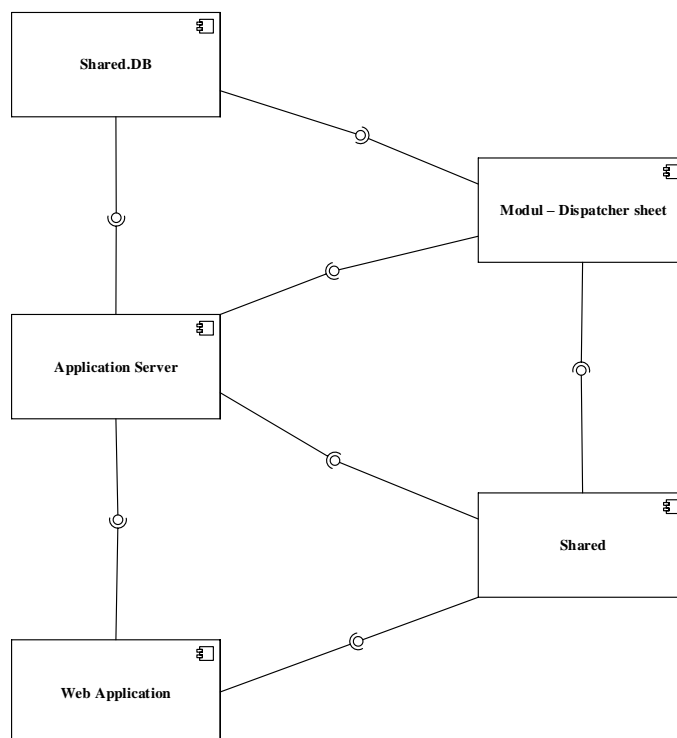
Produkt je vyvíjen v týmu několika studentů. Každý student má za úkol navrhnout a implementovat přidělený modul. V této práci se budeme zabývat modulem dispečerské plachty a ostatními záležitostmi v rámci modulu.

4.1 Architektura projektu

Aplikace je vyvíjena na serverové straně programovacím jazykem C# a jako architektonický vzor slouží MVC. Na klientské straně je používán javascriptový framework jQuery a ajaxové komponenty DevExpress. Vzor MVC neřeší komunikaci s databází, a proto je použita WCF služba [12], která se stará o komunikaci s dalšími knihovnami a přístup k databázi. V projektu se nachází knihovny:

- *Web application* - jedná se o hlavní komponentu aplikace, kde se nachází webová aplikace postavená na asp.net mvc projektu.
- *Application Server* - zajišťuje běh WCF služby a komunikaci s ostatními moduly a knihovnami.
- *Shared* - úložiště tříd a výčtových typů, které se přenášejí na aplikační vrstvě tzv. kontrakty. [13]. Nachází se zde třída Mapper, která přemapovává databázové modely na aplikační objekty.
- *Shared.DB* - je zde uchován databázový model a třídy, které přistupují k databázi pomocí Entity Frameworku.
- *Modul - Dispatcher sheet* - knihovna pro modul dispečerskou plachtu. Je zde uložena logika pro modul.

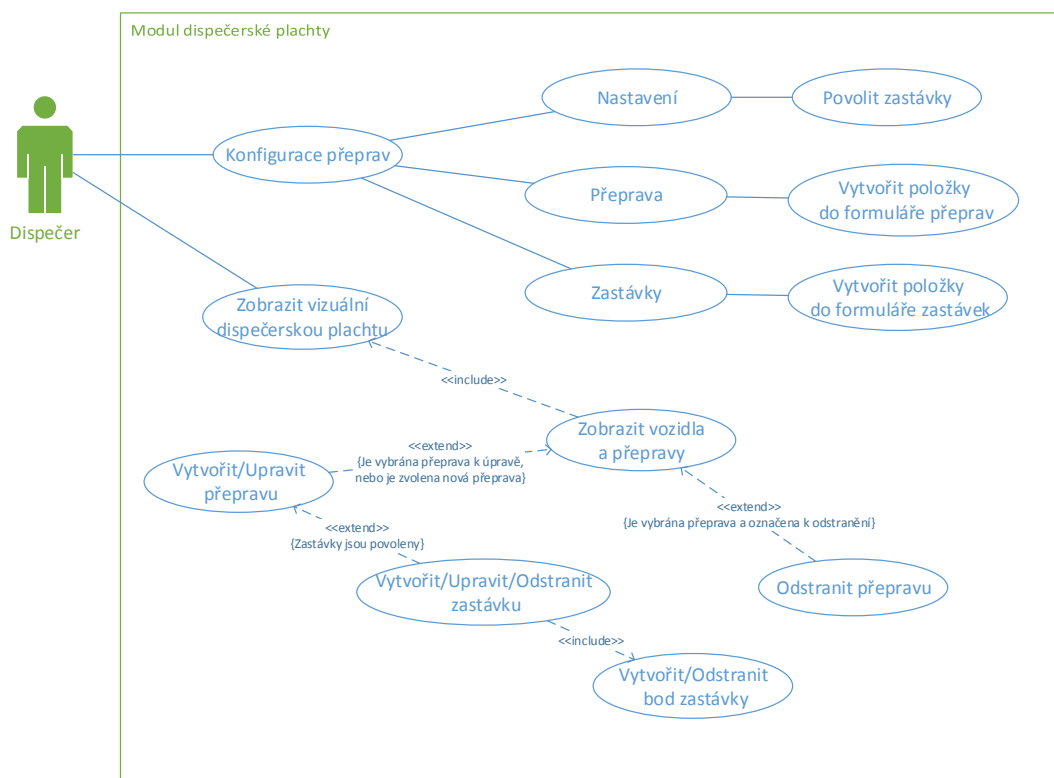
Závislosti mezi jednotlivými knihovnami můžeme vidět na Diagramu komponent v obrázku 6.



Obrázek 6: Diagram komponent

4.2 Návrh modulu

Nově navrhovaný modul dispečerské plachty se bude odvíjet od modulu stávající aplikace smartTDM od firmy Gx Solutions. Modul dispečerské plachty je rozdělen do několika částí. První část obsahuje nastavení modulu, kde si může zákazník vytvořit potřebné položky (vstupy), které budou obsaženy ve formulářích. V další části se na základě nastavení od uživatele vygenerují formuláře pro přepravu a zastávky. Poslední částí modulu je speciální uživatelské rozhraní (upravený kalendář), které slouží pro zobrazení vytvořených přeprav a jednoduchou manipulaci s nimi. Jedná se o funkce drag and drop, editace, mazání. Přepravy bude možné filtrovat na základě přiřazených vozidel. Pro lepší porozumění modulu slouží diagram případu užití viz obr. 7.



Obrázek 7: Příklad užití

4.2.1 Konfigurace přeprav

První část modulu bude sloužit jako nastavení modulu dispečerské plachty. Zákazník si v této části vytvoří položky, podle kterých se později vygenerují formuláře pro přepravu a zastávky v přepravě. Konfigurace přeprav se skládá ze tří záložek:

1. Záložka Nastavení obsahuje obecné nastavení pro celý modul dispečerské plachty. Bude se jednat o jednoduchý formulář s jediným atributem - použít zastávky. Na základě tohoto atributu si zákazník zvolí, zda bude chtít používat u přepravy zastávky. Pokud zákazník poprvé vstoupí do části konfigurace přeprav, v záložce bude defaultně zvolena volba Použít zastávky. Nastavení se bude ukládat podle identifikátoru zákazníka (firmy) a je nadřazené dalším záložkám. Návrh uživatelského rozhraní můžeme vidět na obrázku 8.

The image shows a web interface with three tabs: 'Nastavení', 'Přeprava', and 'Zastávka'. The 'Nastavení' tab is active. Inside this tab, there is a checkbox labeled 'Použít zastávky:' which is checked. Below the checkbox is a button labeled 'Uložit'.

Obrázek 8: Návrh konfigurace přeprav - nastavení

2. Další záložka se bude jmenovat Přeprava. V této sekci bude moci zákazník vytvářet, editovat nebo mazat položky (vstupy), na základě kterých se budou generovat vstupy do formuláře pro přepravu. Pro jednoduchou manipulaci se záznamy bude sloužit tabulkový formulář. Při vytváření nové položky může zákazník nastavit několik hodnot :

- Aktivní - půjde o zaškrťovací položku, kde si zákazník u dané položky zvolí, zda se bude ve formuláři zobrazovat.
- Typ - bude se jednat o výčtový typ, který bude implementovaný jako enum. Bude určovat základní typ položek.
 - Binární
 - Číselný
 - Časový
 - Textový
 - Objektový
- Popis - popis položky se bude zobrazovat ve formuláři jako popis k danému vstupu.
- Formát - výčtový typ bude reagovat na sloupec Typ a na základě vybraného typu se zobrazí správný výčtový typ s podrobnými formáty pro položku.
 - (a) Binární
 - i. Ano
 - ii. Ne
 - (b) Číselný
 - i. desetinné číslo
 - ii. celé číslo
 - iii. procenta

- (c) Časový
 - i. hodiny např. 12:31
 - ii. datum např. 12.12.2015
- (d) Textový
 - i. řetězec
- (e) Objektový
 - i. číselník
 - ii. osoby
 - iii. pobočky
 - iv. uživatelé
 - v. zařízení
 - vi. řidiči
- Ukázat v tabulce - zaškrtnutí vstup na základě kterého se určí, zda se daná hodnota položky má zobrazit v seznamu přeprav.
- Odeslat do terminálu řidiči - oznamuje systému, zda u vytvořené přepravy odeslat vyplněnou hodnotu do terminálu řidiči.

Nastavení		Přeprava	Zastávka				
	Aktivní	Typ	Popis	Formát	Ukázat v tabulce	Odeslat do terminálu	
Save Clear	<input checked="" type="checkbox"/>	Typ <input type="text"/>	<input type="text"/>	Formát <input type="text"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Obrázek 9: Wireframe - konfigurace přeprav

U již vytvořených položek bude možnost určovat jejich pořadí v seznamu. Podle nastaveného pořadí se položky budou generovat postupně do formuláře.

3. Poslední záložka Zastávky bude představovat totožné nastavení. Odlišná funkčnost nastavení zastávek od přeprav nastane až v okamžiku zobrazení zastávek. Formulář pro zastávky bude realizovaný pomocí tabulkového formuláře, jelikož přeprava může obsahovat několik zastávek. Veškeré položky vytvořené v této sekci budou obsaženy v jednom řádku tabulky.

4.2.2 Přeprava

Bude se jednat o důležitou část modulu. Podle vytvořených položek v konfiguraci přeprav a povinných položek se vygeneruje formulář pro přepravu. Tenhle formulář bude základem pro celý proces vytvoření přepravy. Pro dynamické generování položek do formuláře bude potřeba navrhnout funkčnost pro několik scénářů.

1. Vytvoření nové přepravy

Na základě aktuálně vytvořených položek v konfiguraci přeprav se vygenerují formuláře.

2. Úprava již existující přepravy

Je potřeba si dát pozor, aby se položky, které byly dříve vyplněny v přepravě, shodovaly s aktuální konfigurací přeprav.

- V konfiguraci přeprav byly odstraněny položky a upravovaná přeprava zobrazí jen hodnoty aktuálních položek podle konfigurace přeprav.
- V konfiguraci přeprav byly přidány nové položky a upravovaná přeprava má zobrazeny aktuální hodnoty nastavených položek a prázdné hodnoty nově nastavených položek.

U přeprav mohou nastat zmiňované scénáře, které bude potřeba respektovat u implementace algoritmu pro generování položek. Z důvodu pozdějšího napojení na další části modulu musí přeprava obsahovat některé povinné atributy, které se budou mapovat na vizuální dispečerskou plachtu. Jedná se o položky:

- Název
- Popis
- Začátek přepravy
- Konec přepravy
- Vozidlo

Tyto položky jsou povinné a zákazník je musí vždy vyplnit. Po povinných položkách se vygenerují další položky na základě konfigurace přeprav (obr. 10).

Pokud zákazník vyplní položky a zvolí možnost uložit, formulář se přepne do další části, kde se zobrazí další část formuláře. V dalším kroku si zákazník vytvoří zastávky a u každé z nich může určit průjezdné body, které označí na mapě pod formulářem (obr. 11). Zákazník bude moci vytvořenou přepravu stornovat pomocí tlačítka Stornovat, které se nachází vedle tlačítka Uložit. Každá přeprava v sobě uchovává informaci o svém stavu, který si nemůže zákazník zvolit ručně. Stav se odvíjí od vykonané akce na přepravě. Bude se jednat o stavy:

- Vytvořená - stav se nastaví automaticky po vytvoření přepravy
- Stornovaná - stav se nastaví po stornování přepravy

Nová přeprava

Název

Popis

Začátek

▼

Konec

▼

Voždlo

▼

Zákazník

▼

Cena

Uložit

Obrázek 10: Wireframe - nová přeprava

Nová přeprava

Název

Popis

Začátek

▼

Konec

▼

Voždlo

▼

Zákazník

▼

Cena

Uložit

Stomovat

Zastávky

Zákazník

Save

Clear

▼

Body

Bod

Save

Clear

Obrázek 11: Wireframe - zastávky

4.2.3 Vizuální dispečerská plachta

Bude se jednat o speciální uživatelské rozhraní (upravený kalendář), kterému se odborně říká vizuální dispečerská plachta. Bude mít na starost vizuální zobrazení přeprav. Na levé straně kalendáře se budou zobrazovat vybraná vozidla na základě filtrovacího panelu. Přepravy budou zobrazeny v kalendáři na pozicích podle přiřazeného vozidla. Kalendář bude podporovat funkce:

- drag and drop - možnost označit a přesunout přepravu
- upravit - přesměrovat označenou přepravu do editačního formuláře a možnost ji editovat
- smazat - odstranit přepravu přímo na vizuální dispečerské plachtě

Po označení vybraného dne v kalendáři, se zobrazí jednoduchá tabulka, která bude obsahovat informace o přepravách ve zvolený den. Tabulka se bude skládat ze sloupců nastavených v konfiguraci přeprav.

4.3 Databázová vrstva modulu

Požadavky kladené na návrh databázové struktury vznikly na základě návrhu modulu a požadavků od vedení firmy. Bylo potřeba do nově vznikající databáze naimplementovat a provázat mezi sebou tabulky u nově tvořeného modulu. Prvním krokem bude vytvoření tabulky pro konfiguraci přeprav. Na základě těchto tabulek musíme vytvořit tabulky pro přepravu, zastávky, body zastávek a dynamické položky patřící k přepravě. Tabulky vznikající pro tento modul nesou prefix *Transport* pro snadnější orientaci v databázi.

4.3.1 Konfigurace přeprav

Nyní přejdeme k jednotlivým požadavkům pro správnou funkčnost částí modulu. Musíme vytvořit strukturu, která bude uchovávat kompletní konfiguraci přeprav. U konfigurace přeprav musíme počítat s třemi tabulkami.

První tabulka bude uchovávat konfiguraci přeprav - nastavení (obr. 8). je nutné vytvořit jedinečný klíč pro každý záznam. Nastavení je vázáno na zákazníka, proto musí uchovávat jeho identifikační klíč. Dále musí obsahovat atribut, zda daný zákazník bude chtít použít zastávky. Tabulka nese název *TransportConfig*.

Další tabulka bude obsahovat nastavení pro přepravu. Bude se jednat o záznamy vytvořených položek sloužících jako vstupní položky ve formuláři přeprava. Každá vytvořená položka musí obsahovat jedinečný klíč, který se bude dále používat při generování formuláře pro přepravu. V záznamu atributy jako Aktivní, Zobrazit v tabulce nebo Odeslat do terminálu budou reprezentovány boolean hodnotou, kde stačí, abychom věděli, zda je výrok pravdivý, či ne. Atributy jako Typ a Formát Typu budou reprezentovány v systému jako výčtové typy a nebude potřeba pro ně samostatné tabulky. U vybraných hodnot stačí uložit jejich identifikační klíč, pomocí kterého se vybere určitá položka z

výčtového typu. Popis dané položky bude reprezentován jako řetězec a pozice položky bude reprezentována jako celé číslo. Tabulka nese název *TransportConfigData*.

I když se u zastávek jedná o totožné nastavení jako u přeprav, nebudou data uložena v jedné tabulce. Pro zastávky bude vytvořena samostatná tabulka s odlišným názvem, který bude obsahovat totožné atributy jako tabulka přeprav. Tabulka se jmenuje *TransportStopConfigData*. Podrobně rozepsané atributy tabulek můžeme nalézt v příloze B.

Lineární zápis entit

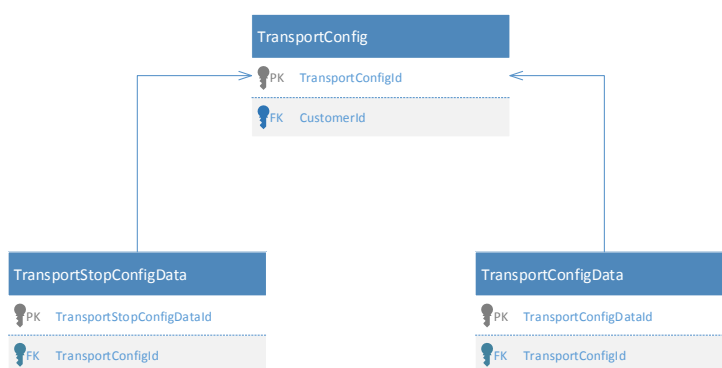
- TransportConfig (**TransportConfigId**, CustomerId, UseStop)
- TransportConfigData (**TransportConfigDataId**, TransportConfigId, isActive, Type, DescriptionType, FormatType, ShowInTable, SendToTerminal, Position)
- TransportStopConfigData (**TransportStopConfigDataId**, TransportConfigId, isActive, Type, DescriptionType, FormatType, ShowInTable, SendToTerminal, Position)

Lineární zápis vztahů

- TransportStopConfigData-TransportConfig (TransportStopConfigData, TransportConfig) 1:1
- TransportConfigData-TransportConfig (TransportConfigData, TransportConfig) 1:1

Entity Relationship Diagram

Vztahy mezi jednotlivými tabulkami jsou zobrazeny v následujícím ER diagramu. Pro jednoduchost jsou zobrazeny jen primární a cizí klíče. Vztahy odpovídají vztahům mezi jednotlivými třídami, které jsou popsány v kapitole 5.



Obrázek 12: ER diagram - konfigurace přeprav

4.3.2 Přeprava

Struktura u přepravy bude o něco komplikovanější, jelikož kromě povinných položek, obsahuje také formulář položky vytvořené zákazníkem. Tyto položky budou muset být uloženy ve speciální tabulce.

Tabulka pro samotnou přepravu musí obsahovat jedinečný klíč, pod kterým se bude moci přeprava později editovat, nebo provádět další události. Jak již bylo zmíněno v návrhu, přeprava musí obsahovat několik povinných atributů, a to z důvodu propojení s další částí modulu. Jedná se o atributy Název a Popis, které budou reprezentovány jako řetězce. Začátek a konec přepravy budou uloženy jako datum. Stav přepravy bude odkazovat identifikačním klíčem na výčtový typ a vozidlo bude odkazovat pomocí cizího klíče na konkrétní vozidlo u přepravy.

Přeprava obsahuje několik napevno daných atributů, které se později budou mapovat na vizuální dispečerskou plachtu. Atributy Název, Popis budou reprezentovat řetězec, Data od-do budou reprezentovány jako datum a Vozidlo bude obsahovat primární klíč vybraného vozidla, čili půjde o celé číslo. Dále je tam atribut status, který bude reprezentovat status dané přepravy. Bude se jednat o Id na výčtový typ stavů.

Veškeré hodnoty položek nastavené v konfiguraci přeprav, přesněji v záložce Přeprava, budou uloženy v samostatné tabulce. Tabulka bude obsahovat konkrétní hodnotu položky, která bude v databázi reprezentována jako řetězec a v systému se bude měnit na základě přiřazeného formátu. Pro rozpoznání formátu k hodnotě musí záznam uchovávat cizí klíč na konkrétní záznam tabulky *TransportConfigData*, který reprezentuje vytvořenou položku a její formát. Abychom mohli hodnoty zpětně namapovat k dané přepravě, musí záznam uchovávat identifikátor na přepravu.

Zastávky nemají žádné povinné položky jako přeprava. Proto celý záznam závisí na počtu a typu položek vytvořených v konfiguraci přeprav - zastávky. Problém nastává, pokud potřebujeme zpětně namapovat, které hodnoty patří do kterého řádku. Abychom tomuto problému předešli, musíme vytvořit tabulku, jejíž identifikační klíč bude reprezentovat id řádku formuláře zastávek. Veškeré hodnoty v tomto řádku ponesou id řádku pro zpětné namapování na konkrétní řádek. Tabulka se bude jmenovat *TransportStopDataRow*, a jak jsme už zmiňovali, bude obsahovat identifikační klíč a cizí klíč na přepravu.

Nyní můžeme vytvořit tabulku reprezentující konkrétní hodnoty zastávek. Musí obsahovat identifikační klíč pro pozdější reprezentaci záznamu a odkaz na záznam tabulky *TransportStopConfigData*, kde se uchovávají vytvořené položky v konfiguraci přeprav - zastávky. Znovu uložíme hodnotu jako řetězec a budeme ji mapovat na konkrétní formát podle reference na položku z konfigurace přeprav - zastávky. Poslední atribut bude reprezentovat odkaz na záznam rodičovské tabulky *TransportStopDataRow* a určuje řádek, ve kterém se má hodnota zobrazit.

Poslední tabulka reprezentuje mapové body k dané zastávce. Mapový bod bude vyjádřen řetězcem, jelikož se bude jednat o název konkrétního místa (město, vesnice, ulice a jiné). Bude obsahovat cizí klíč na záznam *TransportStopDataRow*, který reprezentuje identifikační klíč řádku zastávky. Podrobně rozepsané atributy tabulek můžeme nalézt v příloze B.

Lineární zápis entit

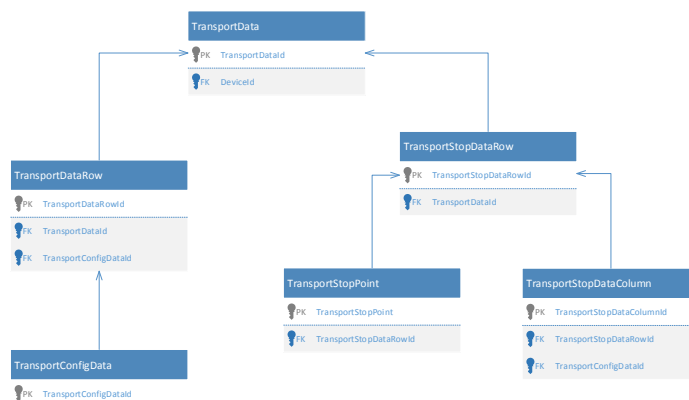
- TransportData (**TransportDataId**, Status, Since, Until, Subject, Description, DeviceId)
- TransportDataRow (**TransportDataRowId**, TransportConfigDataId, TransportDataId, Value)
- TransportStopDataRow (**TransportStopDataRowId**, TransportDataId)
- TransportStopDataColumn (**TransportStopDataColumnId**, TransportStopConfigDataId, TransportStopDataRowId, Value)
- TransportStopPoint (**TransportStopPointId**, TransportStopDataRowId, Title)

Lineární zápis vztahů

- TransportDataRow-TransportData (TransportDataRow, TransportData) N:1
- TransportStopPoint-TransportStopDataRow (TransportStopPoint, TransportStopDataRow) N:1
- TransportStopDataColumn-TransportStopDataRow (TransportStopDataColumn, TransportStopDataRow) N:1
- TransportStopDataRow-TransportData (TransportStopDataRow, TransportData) N:1

Entity Relationship Diagram

Vztahy mezi jednotlivými tabulkami jsou zobrazeny v následujícím ER diagramu. Pro jednoduchost jsou zobrazeny jen primární a cizí klíče. Vztahy odpovídají vztahům mezi jednotlivými třídami, které jsou popsány v kapitole 5.



Obrázek 13: ER diagram - přeprava

5 Implementace

5.1 Výkonné jádro modulu

Modul dispečerská plachta je rozdělen na tři části. V první části je potřeba vytvořit konfiguraci přeprav, ve které si zákazník vytvoří položky a jejich typy, které se budou zobrazovat při vytvoření nové přepravy. V další části se na základě konfigurace přeprav vygenerují požadované formuláře a vyplněné údaje uložíme do databáze. V poslední části zobrazíme uložené přepravy na speciálním uživatelském rozhraní, kde s nimi zákazník může nadále pracovat.

5.1.1 Konfigurace přeprav

Jde o důležitou část modulu, ve které si zákazník vytvoří položky, které budou generovány ve formulářích. Konfigurace přeprav a konfigurace zastávek mají totožné atributy, ale jsou uloženy v odlišných tabulkách. Konfigurace přeprav se dále dělí na tři části:

1. Obecné nastavení konfigurace - jedná se o základní nastavení celé konfigurace přeprav, jak jsme mohli vidět v návrhu na obrázku 8.

Konfigurace (třída - ConfigurationTransports.cs)

- Id - jednoznačný identifikátor konfigurace.
- UseStop - proměnná nabývá hodnot true/false, které dále určují, zda zákazník bude používat zastávky v přepravě.
- CustomerId - identifikační číslo, které odkazuje na konkrétního zákazníka.
- TransportRows - obsahuje seznam vytvořených položek pro zastávky.
- TransportStopRows - obsahuje seznam vytvořených položek pro přepravy.

2. Přeprava - nastavení položek pro přepravy. Je vázané na obecnou konfiguraci přeprav.

Konfigurace - přeprava (třída - ConfigurationTransportsColumnsData.cs)

- Id - jednoznačný identifikátor položky.
- ConfigurationTransportsId - odkaz na obecnou konfiguraci přeprav.
- isActive - určuje, zda je položka aktivní.
- Type - výčtový typ, ve kterém jsou uloženy typy pro položku (řetězec, číslo, datum).
- DescriptionType - název pro danou položku.
- FormatType - specifický formát položky. Jedná se o výčtový typ. Na základě zvoleného Typu může zákazník zvolit formát daného typu, např. datum - dd/MM/yyyy nebo čas - HH:mm.
- ShowInTable - boolean hodnota určí, zda se má položka zobrazit v tabulce Seznam přeprav.

- SendToTerminal - boolean hodnota určí, zda se hodnota dané položky odešle do terminálu.
 - Position - číselná hodnota určuje pořadí v tabulce a pozdější zobrazení ve formuláři při vytváření přepravy.
3. Zastávky - nastavení položek pro zastávky. Je aktivní pokud, je hodnota UseStop true v obecné konfiguraci přeprav.

Konfigurace - zastávky (třída - ConfigurationTransportsColumnsData.cs)

- Id - jednoznačný identifikátor položky .
- ConfigurationTransportsId - odkaz na obecnou konfiguraci přeprav.
- isActive - určuje, zda je položka aktivní.
- Type - výčtový typ, ve kterém jsou uloženy typy (řetězec, číslo, datum).
- DescriptionType - název položky.
- FormatType - specifický formát položky. Jedná se o výčtový typ. Na základě zvoleného Typu může zákazník zvolit formát daného typu, např. datum - dd/MM/yyyy nebo čas - HH:mm.
- ShowInTable - boolean hodnota určí, zda se má položka zobrazit v tabulce Seznam přeprav.
- SendToTerminal - boolean hodnota určí, zda se hodnota položky odešle do terminálu.
- Position - číselná hodnota určuje pořadí v tabulce a pozdější zobrazení ve formuláři při vytváření přepravy.

Vytvořené uživatelské rozhraní pro konfiguraci přeprav můžete vidět v Příloze A na obr. 20.

5.1.2 Formuláře pro novou přepravu

Na základě informací nastavených v konfiguraci přeprav je potřeba vygenerovat dynamické formuláře pro přepravu a zastávky. Další nutností je vykreslení tabulky sloužící k zadávání průjezdných bodů vybraných z mapy.

V konfiguraci přepravy byly vytvořeny položky, které zákazník chce zobrazit na formuláři. Musíme vytvořit třídu, která bude reprezentovat hodnotu dané položky a odkazovat na položku, pomocí které namapujeme hodnotu na daný formát.

Hodnota položky formuláře - (třída - TransportRowsDynamic.cs)

- Id - jednoznačný identifikátor vstupu.
- TransportFieldId - identifikátor položky v konfiguraci přeprav, která určuje typ, formát a jiné informace o položce.

- Value - řetězec, kde se ukládají data ze vstupu a potom se pomocí odkazu na TransportFieldId naformátují podle navoleného formátu.
- TransportDataId - identifikátor přepravy.

Formulář pro přepravu obsahuje položky, které jsou fixně dány (Status, StartTime, EndTime, Subject, Description, ResourceId), a položky, které se vygenerují na základě nastavení v konfiguraci přeprav - přeprava. Jedná se o položky, které jsou součástí třídy.

Přeprava - (třída - TransportDataDynamic)

- Id - jednoznačný identifikátor přepravy.
- Status - výčtový typ, určuje stav přepravy - vytvořená, stornovaná.
- StartTime - datum určující začátek přepravy.
- EndTime - datum určující konec přepravy.
- Subject - řetězec nese název přepravy.
- Description - řetězec pro poznámku.
- ResourceId - odkaz na vozidlo, kterému je přeprava určena.
- TransportRows - obsahuje seznam hodnot položek, které byly vytvořeny v konfiguraci přeprav.

Seznam hodnot položek, který je reprezentován atributem TransportRows, se naplní daty pomocí algoritmu.

```
public List<TransportRowsDynamic> GetTransportDataRowForExistForm(int transportDataId,
    int customerId)
{
    List<TransportDataRow> result = new List<TransportDataRow>();
    List<TransportConfigData> sloupceKonfigurace = new List<TransportConfigData>();

    //Ziska aktualni konfiguraci podle zakaznika
    sloupceKonfigurace = Database.GetTransportConfigDataByCustomerId(customerId)
    ;

    for (int i = 0; i < sloupceKonfigurace.Count(); i++)
    {
        var item = sloupceKonfigurace[i];
        // Zjistí , zda k polozce z konfigurace existuje hodnota. Pokud ano hodnota se ulozi
        if (Database.CheckIfTransportDataRowExist(transportDataId, item.
            TransportConfigDataId))
        {
            var transportDataRow = Database.
                GetTransportDataRowByTransportIdAndTransportConfigDataId(
                    transportDataId, item.TransportConfigDataId);
```

```

        result.Add(transportDataRow);
    }
    // Pokud hodnota k polozce v databazi neni, vytvori se prazdna hodnota
    else
    {
        if (item.Type == (int)SoftInputNum.Chronologic)
            result.Add(new TransportDataRow { TransportDataID =
                transportDataId, TransportConfigDataID = item.
                TransportConfigDataID, Value = DateTime.Now.ToString() });
        else
            result.Add(new TransportDataRow { TransportDataID =
                transportDataId, TransportConfigDataID = item.
                TransportConfigDataID, Value = "" });
    }
}

// Prematuje
List<TransportRowsDynamic> list = new List<TransportRowsDynamic>();
list = TransportDataRowMap(result);

// Vratí seznam hodnot k daným položkám
return list ;
}

```

Výpis 1: Algoritmus pro získání hodnot k položkám

K přepravě se dále váže tabulkový formulář pro zastávky. Tento formulář je trochu složitější, jelikož veškeré položky tvoří jeden řádek v tabulce. Musíme vytvořit třídu, která reprezentuje řádky tabulky pro zastávky a jsou na ni namapovány veškeré hodnoty, které spadají do daného řádku.

Řádek zastávky (třída - **TransportStopDataRowData.cs**)

- **TransportStopDataRowDataId** - jednoznačný identifikátor řádku v tabulce zastávek
- **TransportDataId** - odkazuje na přepravu

Reprezentuje objekt pro uložení informací o řádku a hodnotě sloupce.

Sloupec řádku (třída - **TransportStopDataColumnData.cs**)

- **TransportStopDataColumnId** - jednoznačný identifikátor sloupce v tabulce zastávek.
- **TransportField** - reference na konkrétní nastavení v konfiguraci přeprav - zastávka.
- **Value** - hodnota daného sloupce.
- **Row** - identifikátor odkazující na rodičovskou tabulku, která reprezentuje řádek tabulky.

Pro namapování hodnot zastávek k danému řádku musí být vytvořen objekt, který ne-definuje konkrétní datový typ.

Reprezentuje řádkový záznam (třída- RowData.cs)

- Values - obsahuje seznam nedefinovaných objektů

```

public List<RowData> RowDataStops(int transportId, int customerId)
{
    List<RowData> result = new List<RowData>();
    // Ziska aktualni polozky z konfigurace preprav – zastavky
    var columnsConfig = Database.GetTransportStopConfigDataByCustomerId(
        customerId);
    // Ziska seznam radku pro tabulku
    List<TransportStopDataRow> transportRows = Database.
        GetTransportStopDataRowByTransportDataId(transportId);
    // Projde vsechny nastavenne polozky
    for (int i = 0; i < transportRows.Count(); i++)
    {
        var transportRow = transportRows[i];

        RowData row = new RowData();
        // Prvni hodnota tabulky je ID
        row.Values.Add(transportRow.TransportStopDataRowID);
        for (int y = 0; y < columnsConfig.Count(); y++)
        {
            var columnConfig = columnsConfig[y];
            // Zjistí zda dana polozka ma ulozenou hodnotu v databazi, a pokud ano tak
            // ulozi hodnotu
            var checker = Database.
                CheckIfExistTransportStopDataColumnByTransportStopConfigDataIdandRowID
                (columnConfig.TransportStopConfigDataID, transportRow.
                    TransportStopDataRowID);
            if (checker == true)
            {
                var data = Database.
                    GetTransportStopDataColumnByTransportStopConfigDataIdandRowID
                    (columnConfig.TransportStopConfigDataID, transportRow.
                        TransportStopDataRowID);
                row.Values.Add(data.Value);
            }
            // Pokud polozka nema hodnotu, prida prazdny retezec
            else
            {
                row.Values.Add("");
            }
        }
        // Vsechny hodnoty ulozi do radku
        result.Add(row);
    }
}

```

```

        // Vratí seznam radku
        return result ;
    }

```

Výpis 2: Algoritmus pro namapování hodnot na konkrétní řádek zastávek

Vytvořené uživatelské rozhraní pro přepravu můžete vidět v Příloze A obr. 21.

5.1.3 Vizuální dispečerská plachta

Vizuální dispečerská plachta představuje vizuální zobrazení vytvořených přeprav. Dále můžeme v dispečerské plachtě s přepravami manipulovat pomocí funkcí editace, drag and drop, kopírování. DevExpress nabízí komponentu scheduler [15], která se chová jako kalendář.

Dalším krokem u komponenty je změna defaultního formuláře na námi vytvořený formulář. Je potřeba změnit kontextové menu na komponentě. Při přesměrování do formuláře je potřeba rozeznat, o jakou akci se jedná (Vytvořit novou přepravu, nebo ji editovat). Pro tyto účely nám slouží název položky z kontextového menu (OpenAppointment, NewAppointment), podle kterého na základě jména zobrazíme požadovaný formulář.

```

function onMenuClick(s, e) {
    if (e.item.GetItemCount() > 0)
        return;
    // Pokud klikne na kontextove menu a vybere moznost zobrazit na mape
    if (e.item.name == "ShowMap") {
        var item = { item: HeaderMenu.GetItemByName("/PrintForms/Index") };
        var menuItemProp = {
            item: {
                name: "/Map/IndexPartial",
                data: { datas: dataFromTilt, state: e.item.name },
                parent: item.item.parent,
                items: []
            }
        };
        onMainMenuItemClick(HeaderMenu, menuItemProp);
    }
    // Smazat prepravu
    else if (e.item.name == "DeleteAppointment") {
        dispatcherScheduler.RaiseCallback("MNUAPT|" + e.item.name);
    }
    // Vytvorit nebo upravit prepravu -> presmerovani na vlastni formular
    else if (e.item.name == "OpenAppointment" || e.item.name == "NewAppointment") {
        var item = { item: HeaderMenu.GetItemByName("/PrintForms/Index") };
        var menuItemProp = {
            item: {
                name: "/Map/IndexPartial",
                data: { datas: dataFromTilt, state: e.item.name, device: resourceId },
                parent: item.item.parent,
                items: []
            }
        };
        onMainMenuItemClick(HeaderMenu, menuItemProp);
    }
}

```

```

    }
}

```

Výpis 3: Algoritmus pro namapování hodnot na konkrétní řádek zastávek

Pro zobrazení přeprav na komponentě je potřeba poslat správná data. Komponenta vyžaduje data, podle kterých zobrazí přepravu na správném místě. Jedná se o totožná data, která jsou povinná při vytváření přepravy (výpis kódu 4). Stejné nastavení musíme provést pro správné zobrazení vozidel (výpis kódu 5).

Resource - třída, která reprezentuje vozidla (třída - TransportResource.cs)

- ResourceId - jednoznačný identifikátor vozidla
- ResourceName - název vozidla

```

public static MVCxAppointmentStorage CreateStorage()
{
    MVCxAppointmentStorage appStorage =
        new MVCxAppointmentStorage();
    appStorage.Mappings.AppointmentId = "Id";
    appStorage.Mappings.Start = "StartTime";
    appStorage.Mappings.End = "EndTime";
    appStorage.Mappings.Subject = "Subject";
    appStorage.Mappings.Description = "Description";
    appStorage.Mappings.ResourceId = "ResourceID";
    return appStorage;
}

```

Výpis 4: Metoda, která nastaví mapování třídy pro přepravu na položku komponenty

```

public static MVCxResourceStorage CreateDefaultResourceStorage()
{
    MVCxResourceStorage resourceStorage = new MVCxResourceStorage();
    resourceStorage.Mappings.ResourceId = "ResourceID";
    resourceStorage.Mappings.Caption = "ResourceName";
    return resourceStorage;
}

```

Výpis 5: Metoda, která nastaví mapování třídy vozidla (resource) na zdroj pro komponentu

Pro správnou funkci události drag and drop komponenta poskytuje rozšířené metody pro rychlou editaci, vložení a mazání. Abychom kvůli jednoduché editaci nemuseli otevírat celý formulář s přepravou, využijeme těchto metod. Jde o statické metody GetAppointmentsToInsert, GetAppointmentsToUpdate, GetAppointmentsToRemove, které vrací seznam přeprav. Jejich parametry jsou:

- Název komponenty
- Seznam přeprav

- Seznam vozidel
- Mapování dat z přeprav na komponentu (výpis kódu 4).
- Mapování dat vozidel na komponentu (výpis kódu 5).

```
// Zjistí změny na preprave
var updatedAppt = SchedulerExtension.GetAppointmentsToUpdate<TransportDataDynamic>("
    dispatcherScheduler", client.GetTransportData(),
    GetListDevices(), AppointmentMappingStorage.DefaultAppointmentStorage,
    AppointmentMappingStorage.DefaultResourceStorage);
foreach (var appt in updatedAppt)
{
    UpdateAppointment(appt);
}
```

Výpis 6: Metoda získá seznam upravených přeprav

V poslední části je potřeba zobrazit seznam přeprav podle zvoleného dne a vozidla na komponentě. Využijeme javascriptovou funkci `OnSelectionChangedItem`, která se vyvolá pokaždé, když zákazník klikne na komponentu. Do funkce se předávají parametry, které nesou informace o vozidle a datu. Na základě těchto informací pošleme callback na controller, kde se zavolá příslušná metoda. Ta vrátí seznam přeprav odpovídající našim parametrům (výpis kódu 7).

```
[HttpPost]
public ActionResult DetailAppointment(DateTime date, string resourceId)
{
    using (MainServiceClient client = new MainServiceClient())
    {
        if (resourceId != "null")
            return PartialView("DetailsPartial", client.GetTransportDataByDevicAndDate(
                date, Int32.Parse(resourceId)));
        else
            return PartialView("DetailsPartial", new List<TransportDataDynamic>());
    }
}
```

Výpis 7: Metoda vracející seznam přeprav podle data a vozidla

Vytvořené uživatelské rozhraní pro vizuální dispečerskou plachtu můžete vidět v Příloze A na obr. 22.

5.2 Testování a refaktoring modulu

Samotné testování je náročný proces, který zabírá nemalou část vývoje. Přesto po testování není zaručené, že v softwaru nejsou další chyby. Z tohoto důvodu se provádí několik částí testování.

V rámci tvorby modulu dispečerské plachty, bylo průběžně prováděno tzv. Manuální testování, kdy máme za úkol pomocí klikání v rámci modulu narazit v ideálním

případě nenarazit na chybu, která by tam být neměla. Nalezené chyby při manuálním testování jsme se pokoušeli opravit co nejrychleji, aby chyba neměla vliv na správný chod modulu. Před dokončováním implementace modulu jsme prováděli performance testy, které jsou přímo zabudované ve vývojovém prostředí Visual Studia. Tyto testy nám ukáží náročnost jednotlivých metod u modulu a na základě refaktoringu jsme optimalizovali nejnáročnější metody a zrychlili odezvu. Průměrně se metody zrychlily o 1/3 svého původního času.

6 Závěr

Cílem práce bylo navrhnout a vyvinout modul dispečerské plachty. Úkolem modulu je veškerá manipulace s přepravou. Bylo potřeba navrhnout a vytvořit speciální nastavení modulu, které určuje povinné položky při vytváření přeprav a zastávek. Na základě tohoto nastavení jsme vygenerovali formulář pro přepravu a uložili ji. Dalším úkolem bylo zobrazení na speciálním uživatelském rozhraní vytvořené přepravy k jednotlivým vozidlům. Při tvorbě speciálního uživatelského rozhraní jsme použili komponentu Scheduler, která má vlastnosti kalendáře, kterou jsme upravili podle našich potřeb. Dispečer může ve vytvořeném rozhraní řídit vytíženost vozidel a jednoduše s přepravami manipulovat. V průběhu vývoje jsme prováděli jednoduché manuální testování aplikace za účelem ověření funkčnosti vytvořeného modulu. V závěrečné fázi vývoje byly prováděny zátěžové testy, které měly za úkol zobrazit rychlost jednotlivých funkcí a na základě těchto výsledků se refaktoroval kód, který vedl ke zvýšení rychlosti modulu.

Vývoj celého systému probíhal v týmu lidí, ve kterém měl každý za úkol implementaci své části aplikace. Vývoj probíhal podle harmonogramu, který byl stanoven na začátku spolupráce. Největší zdržení nastalo počátkem vývoje, kdy bylo potřeba se seznámit s architekturou systému. Jsem rád, že jsem mohl být v týmu lidí, kteří stojí u zrodu nového systému a vyzkoušet si celý proces od návrhu až po testování informačního systému takové velikosti. Z pohledu budoucího vývoje nám byla nabídnuta spolupráce s firmou a možnost systém dále vyvíjet a udržovat.

Tomáš Jurásek

7 Reference

- [1] O firmě. *Gx solutions* [online]. [cit. 2015-02-21]. Dostupné z: <http://www.gxsolutions.cz/o-firme>
- [2] Monitoring kamionů, autobusů, lokomotiv. *GX solutions* [online]. [cit. 2015-04-17]. Dostupné z: <http://www.gxsolutions.cz/monitoring-kamionu-gx-truck-logistic>
- [3] GPS monitoring a telematika ve výrobě a průmyslu. *Gx Solutions* [online]. [cit. 2015-04-17]. Dostupné z: <http://www.gxsolutions.cz/monitoring-techniky-vyroba-prumysl>
- [4] Sledování osobních aut a styl jízdy řidičů. *GX solutions* [online]. [cit. 2015-04-17]. Dostupné z: <http://www.gxsolutions.cz/sledovani-aut/gx-fleet>
- [5] Sledování práce a využití stavebních strojů. *Gx Solutions* [online]. [cit. 2015-04-17]. Dostupné z: <http://www.gxsolutions.cz/monitoring-stroju/gx-build-agro>
- [6] Monitoring komunální techniky a práce nástaveb. *Gx Solutions* [online]. [cit. 2015-04-17]. Dostupné z: <http://www.gxsolutions.cz/sledovani-komunalni-techniky/gx-municipal>
- [7] Přehled řešení. *GX solutions* [online]. [cit. 2015-03-15]. Dostupné z: <http://www.gxsolutions.cz/gps-monitoring/telematika/fleet-management>
- [8] Smart TDM-monitorovací a komunikační systém. *Gx solutions* [online]. [cit. 2015-02-21]. Dostupné z: <http://www.gxsolutions.cz/software-smart-tdm>
- [9] Produkty - Hi Software Development. *Hi Software Development* [online]. [cit. 2015-02-22]. Dostupné z: <http://www.hisoftware.eu/rubriky/produkty/webdispecink#webdispecink>
- [10] Spedice manuál. *Hi software development* [online]. [cit. 2015-02-22]. Dostupné z: http://www.webdispecink.cz/downloads/Spedice_manual.pdf
- [11] Spedice. *KSH-Data* [online]. [cit. 2015-02-22]. Dostupné z: <http://www.ksh-data.cz/produkty/doprava-3k/moduly/spedice/>
- [12] Úvod do WCF. *Vyvojar.cz* [online]. [cit. 2015-03-02]. Dostupné z: <http://www.vyvojar.cz/Series/2-uvod-do-wcf.aspx>
- [13] WCF - Kontrakty. *Vyvojar.cz* [online]. [cit. 2015-03-02]. Dostupné z: <http://www.vyvojar.cz/Articles/457-wcf-kontrakty.aspx>
- [14] ASP.NET Controls and MVC Extensions. *DevExpress* [online]. [cit. 2015-03-11]. Dostupné z: <https://www.devexpress.com/Products/NET/Controls/ASP/MVC/>

- [15] ASP.NET MVC Scheduler Extension. *DevExpress* [online]. [cit. 2015-03-13]. Dostupné z: <https://demos.devexpress.com/MVCxSchedulerDemos/>
- [16] PATTON, Ron. *Testování softwaru*. Praha: 2002, COMPUTER PRESS. ISBN 80-7226-636-5.
- [17] PIETRIK, Michal. *Automatizované testování webových aplikací*. Brno, 2012. Diplomová práce. MASARYKOVA UNIVERZITA FAKULTA INFORMATIKY.
- [18] Manuální testování softwaru. *Testování softwaru* [online]. [cit. 2015-03-24]. Dostupné z: <http://testovanisoftwaru.cz/manualni-testovani/>

A Dispečerská plachta

Webdispečink - Google Chrome

www.webdispecink.cz/editace_nastav.php?zalozka=dispecer&carid=-1&act=ins

Dispečerská plachta:

Číslo Stav

Vozidlo: Krátký popis

Přípojně vozidlo

Řidič Podrobný popis

jiný řidič

Objednatel

Cena přepravy Kč

Nakládka

Datum od 31 Čas

Vlastní místo

Firma

Kontakt

Kontakt mobil

Ulice

Obec

PSČ

Stát

Vykládka

Datum do 31 Čas

Vlastní místo

Firma

Kontakt

Kontakt mobil

Ulice

Obec

PSČ

Stát

OK Zrušit

Obrázek 14: Webdispečink - Jednoduchá přeprava

« Přepravy

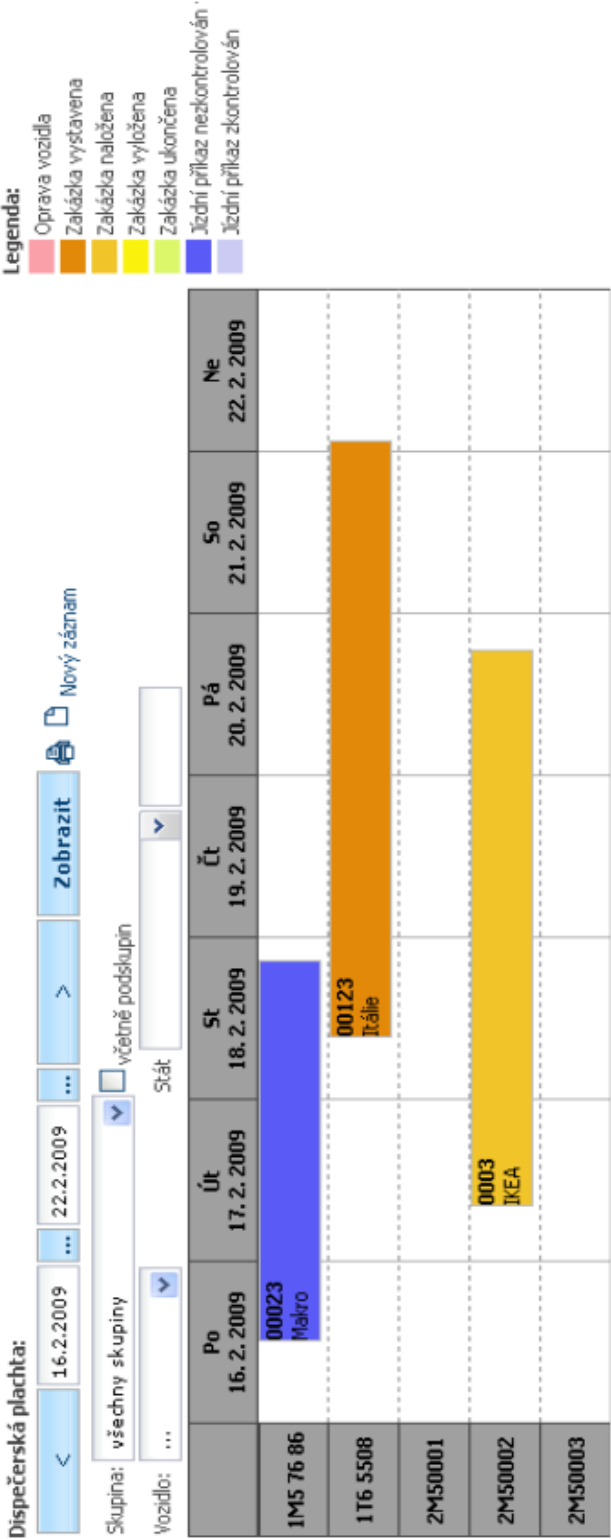
Nová přeprava Zastávky Mapa Alarmy Jízdy Provedené akce Fakturace

Název		Stav	rozpracovaný
Vozidlo	20150222000828	Ridič	...
Odkud		Kam	
Datum od	22.02.2015 12:02	Datum do	22.02.2015 23:59

Poznámka

« Zpět » Další »

Obrázek 15: Webdispečink - Proces vytvoření přepravy



Obrázek 16: Webdispečink - Dispečerská plachta

Seznam přeprav:

Vozidlo:

všechna vozidla

Zobrazit

Nová přeprava

Zobrazit jako dispečerskou plachtu

▼ Filtr

	Vozidlo	Název	Datum od	Datum do	Odkud	Kam	Další zastávka	...	km	Slav	Aktuální poloha	Odjezd	Příjezd
	5M6 7728	2015011...	29.03.201...	29.03.201...	Olomouc, Legionářs...	Opava	Přerov Firma 1	1	110.5	budoucí (12d...			
	5M6 7728	2015011...	29.03.201...	29.03.201...	Olomouc, Legionářs...	Olomouc, Legionářs...	Přerov Firma 1	1	48.43	budoucí (12d...			

Záznamů na stránku: 10 20 50

stránka 1/1 (celkem 2 záznamů)

Obrázek 17: Webdispečink - Seznam přeprav

Zakázka - neuložena

Zakázka
Číslo dokladu: **ZA-000001/1102**
Datum vystavení: **1.7.2011**
Doklad vystavil: **administrátor KSH-Data**

Typ: **Standardní** ☒ Zakázka je kompletní
Číslo obj. zakázky: **2011-0256/00B**

Zákazník: **Partner Logistic a.s.**
Firma: **100023**
Nové sídlo 347
Praha - východ
CZE

Plátce přepravy: **14**

Uložit
Zrušit změny
Rozepsat
Objednávkou
Tisk

Detail | Cena přepravy | Místa nakl. a vykládky | Poznámka k zakázce | Další údaje | Celnice | Související dokumenty

Datum přepr.: **04.07.2011**
Typ plnění: **Tuzemsko**
Cena přepravy: **26 500,00** DPH: **20** Měna: **CZK** Celková cena: **26 500,00 Kč**
Faktura číslo: Částka (Tuz.):

Datum hlášení:
Plánované: **412 km**
Náklad: **Přeprava paletového zboží**
Hmotnost: **0 t** Objem: **26 m³** pal. Cena nákladu:

Podzakázky

Detail - 1 | Místa nakl. a vykládky | Vozidlo | Poznámka k podzakázce | Další údaje | Neshody

Datum přepr.: **04.07.2011**
Objednávka: **00-000036/11000**
Datum hlášení:
Plánované: **285 km**
Plán./skut. km-stažka: **0**
Smluvní cena: **8 500,00**

Čís. řád.	Datum přepravy	Náklad	Objem	m³ obj.	Objednávka	SPZ	Typ plnění
1	4.7.2011	Paletové zboží	12 pal		00-000036/11000	U0 10-10	Tuzemsko
2	5.7.2011	Paletové zboží	14 pal		00-000037/11000	2E1 5028	Tuzemsko

SPZ:

Seznam
Přidat
Kopírovat
Opravit
Smazat
Objednávka
Odeslat zprávu
Odeslat nabídku

Obrázek 18: Doprava 3K - Nová přeprava

Dispečerská plachta, Program Doprava 3K, Licence: KSH-Data s.r.o.

Tyden od: **23.10.2006** do: **29.10.2006**





























Filtra: **<Všechny řady>** Druk: **<Všechny druhy>** TZN: **<Všechny typové značky>**

	143 (pondělí) 23.10.	144 (úterý) 24.10.	145 (středa) 25.10.	146 (čtvrtek) 26.10.	147 (pátek) 27.10.	148 (sobota) 28.10.	149 (neděle) 29.10.
224 8736 1E3 9050	ST-000015/0601 OO-000036/0601 CZE Landkron	Výměna oleje Naplánováno na 24.10.2006	ST-000021/0601 OO-000034/0601 CZE Praha 3 -> GBR London				
U0C 99-88 2E5 2090	ST-000002/0601 OO-000005 Praha 3 -> AUT Salzburg	ST-000036/0602 OO-000034/0600 CZE Hradec -> AUT Salzburg					
U0 10-10 2E4 9050	OV-000008/06000				Technická kontrola Naplánováno na 27.10.2006		
U0J 57-17 2E4 9057	ST-000027/0601 OO-000015/0600 ITA Bobano -> DEU Lindau						
UHK 3268 U0C 11-13	ST-000045/06000 OO-000056/0601 CZE Praha -> ESP Madrid						
1E3 5555	OV-000009/06000 ST-000045/06000 OO-000037/0600 CZE Landkron -> POL Wrocław						
222 3503		ST-000047/06000					
1E3 2580	ST-000044/06000 OO-000005 CZE Lipník	OO-000058/0601 POL Krakow -> CZE					

Nepřidělené zakázky

Doklad	Datum vyst.	Datum přes.	Zákazník	Zákazník město	Zákazník PSC	Zákazník stát	Smluvní cena	Kont. osoba	Vystavil
ZA-000082/06000	24.10.2006	25.10.2006	Novakbau GmbH	Linz	50010	AUT	1 000,00	Karlbauer	Otevřít
ZA-000077/06000	24.10.2006	27.10.2006	Zelezárny Třebíč s.r.o.	Třebíč	25699	CZE	8 200,00	Hradíra	Všechny zak.

Obrázek 19: Doprava 3K - Dispečerská plachta

Přeprava			Zastávka		Konfigurace	
#	Aktivní	Typ	Popis	Formát	Ukázat v tabulce	Odeslat do terminálu
   	<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
   	<input checked="" type="checkbox"/>	Numerický	Císlo	C2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
   	<input checked="" type="checkbox"/>	Řetězcový	Znak	S	<input type="checkbox"/>	<input type="checkbox"/>
   	<input checked="" type="checkbox"/>	Objektový	Osoby	Osoby	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
   	<input checked="" type="checkbox"/>	Řetězcový	Jméno	S	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
   	<input checked="" type="checkbox"/>	Časový	Datum	dd.MM.yyyy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Page 1 of 1 (5 items)    						
Page size: 50						<input type="text"/>

Obrázek 20: Vytvořené uživatelské rozhraní konfigurace přeprav

Online pozice

Dispečerská plachta

Název

Test

Popis

Testovací přeprava

Začátek

10. 4. 2015 2:00

Konec

12. 4. 2015 8:00

Zařízení

Bober M.

Císlo

123,00 Kč

Znak

asd

Osoby

Jurásek Tomáš

Jméno

AAA

Datum

10.04.2015

Uložit

Stomovat

Odeslat do terminálu

Zastávky

#	císlo	Datum	Binární	Jmeno	osob
	2232	10. 4. 2015 0:00:00	<input checked="" type="checkbox"/>	ASDD	Burian2 Jiří

Body k zastávce

#	Popis	Bod
	Ždírec nad Doubravou 14, 582 63 Ždírec nad Doubravou, Česká republika	15,8702363068848, 49,7449518691766

+

-

↑

↓

none

Navteq GX

Open Street

Kováma

Ostrava

Přidat bod zastávky

470

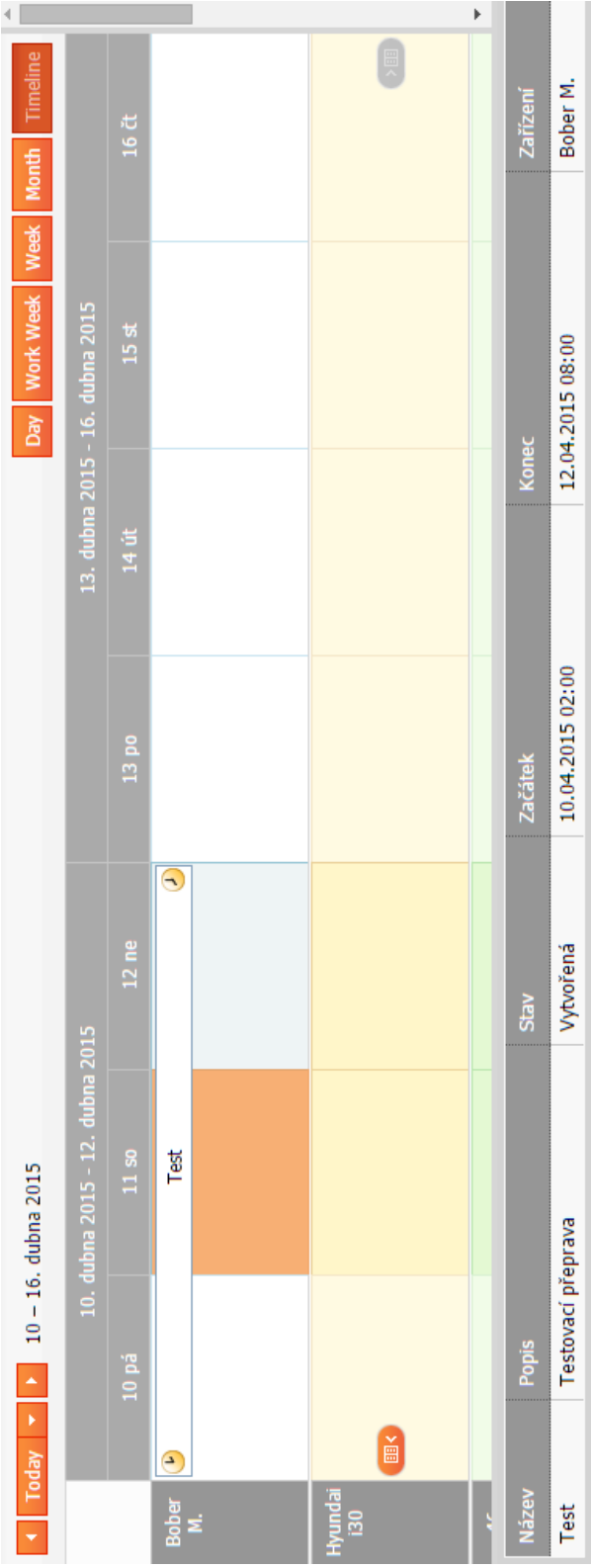
471

472

473

474

Obrázek 21: Vytvořené uživatelské rozhraní pro přepravu



Obrázek 22: Vytvořené uživatelské rozhraní pro vizuální dispečerskou plachtu

B Databáze

Název	Datový typ	Klíč	Null	Index	Popis
TransportConfigId	integer	PK	Ne	Ano	Id nastavení
CustomerId	integer	FK	Ne	Ano	Id zákazníka
UseStop	boolean	Ne	Ne	Ne	použití zastávek

Tabulka 2: Datový slovník tabulky TransportConfig

Název	Datový typ	Klíč	Null	Index	Popis
TransportConfigDataId	integer	PK	Ne	Ano	Id nastavení přeprav
TransportConfigId	integer	FK	Ne	Ano	Id nastavení
isActive	boolean	Ne	Ne	Ne	aktivní položka
Type	integer	Ne	Ne	Ne	Id typu v číselníku
DescriptionType	varchar	Ne	Ne	Ne	název položky
FormatType	integer	Ne	Ne	Ne	Id formátu v číselníku
ShowInTable	boolean	Ne	Ne	Ne	zobrazit v tabulce
SendToTerminal	boolean	Ne	Ne	Ne	odeslat do terminálu
Position	integer	Ne	Ne	Ne	pozice položky

Tabulka 3: Datový slovník tabulky TransportConfigData

Název	Datový typ	Klíč	Null	Index	Popis
TransportConfigDataId	integer	PK	Ne	Ano	Id nastavení přeprav
TransportConfigId	integer	FK	Ne	Ano	Id nastavení
isActive	boolean	Ne	Ne	Ne	aktivní položka
Type	integer	Ne	Ne	Ne	Id typu v číselníku
DescriptionType	varchar	Ne	Ne	Ne	název položky
FormatType	integer	Ne	Ne	Ne	Id formátu v číselníku
ShowInTable	boolean	Ne	Ne	Ne	zobrazit v tabulce
SendToTerminal	boolean	Ne	Ne	Ne	odeslat do terminálu
Position	integer	Ne	Ne	Ne	pozice položky

Tabulka 4: Datový slovník tabulky TransportStopConfigData

Název	Datový typ	Klíč	Null	Index	Popis
TransportDataId	integer	PK	Ne	Ano	Id přepravy
Status	integer	Ne	Ne	Ano	Id stavu v číselníku
Sice	timestamp without time zone	Ne	Ne	Ne	datum začátku přepravy
Until	timestamp without time zone	Ne	Ne	Ne	datum konce přepravy
Subject	varchar	Ne	Ne	Ne	název přepravy
Description	varchar	Ne	Ne	Ne	popis přepravy
DeviceId	integer	FK	Ne	Ano	Id vozidla

Tabulka 5: Datový slovník tabulky TransportData

Název	Datový typ	Klíč	Null	Index	Popis
TransportDataRowId	integer	PK	Ne	Ano	Id hodnoty řádku přepravy
TransportConfigDataId	integer	FK	Ne	Ano	Id nastavení přeprav
TransportDataId	integer	FK	Ne	Ano	Id přepravy
Value	varchar	Ne	Ne	Ne	hodnota řádku

Tabulka 6: Datový slovník tabulky TransportDataRow

Název	Datový typ	Klíč	Null	Index	Popis
TransportStopDataRowId	integer	PK	Ne	Ano	Id řádku zastávky
TransportDataId	integer	FK	Ne	Ano	Id přepravy

Tabulka 7: Datový slovník tabulky TransportStopDataRow

Název	Datový typ	Klíč	Null	Index	Popis
TransportStopDataColumnId	integer	PK	Ne	Ano	Id hodnoty řádku zastávky
TransportStopConfigDataId	integer	FK	Ne	Ano	Id nastavení zastávek
TransportStopDataRowId	integer	FK	Ne	Ano	Id řádku zastávky
Value	varchar	Ne	Ne	Ne	hodnota řádku

Tabulka 8: Datový slovník tabulky TransportStopDataColumn

Název	Datový typ	Klíč	Null	Index	Popis
TransportStopPointId	integer	PK	Ne	Ano	Id bodu zastávky
TransportStopDataRowId	integer	FK	Ne	Ano	Id řádku zastávky
Title	varchar	Ne	Ne	Ne	hodnota řádku

Tabulka 9: Datový slovník tabulky TransportStopPoint